Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

02005419.3

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 02005419.3
Demande no:

Anmeldetag:
Date of filing: 08.03.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Sony Ericsson Mobile Communications AB

221 88 Lund
SUEDE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Security protection for data exchange

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H04M/

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

Security Protection for Data Exchange

Field of the Invention

The present invention relates to transmission protocols for exchanging data between a client and a server. More particularly, the present invention relates to a method for providing authentication and integrity protection when a particular protocol is utilized.

Background of the Invention

Examples of a client as disclosed in this document is portable communication equipment such as mobile telephones, pagers, and communicators, i.e. electronic organizers, smartphones or the like.

In some situations it is desirable to run a device manager through a protocol, such as e.g. SyncML, between a client and a server, to which the client is connected through e.g. a wireless connection. An example of this is when there is a problem with the client and a server of repairer is contacted through the phone. However, the repairer may want to check the authentication of the client before he/she starts repairing the client. Also, the client needs to verify the authentication maid by the repairer to avoid that an unauthorized party, such as a hacker, gets access to the client.

The following security requirements for SyncML-DM have been identified:

- Server authentication
- Client authentication
- Integrity protection
- Confidentiality

Current best practice is to use a combination of transport level and SyncML level security as indicated by table 1.

	SyncML/ Http/X	SyncML/ WSP/X	SyncML/ Obex/Cable or IrDa	SyncML/ Obex/ BT
SyncML layer	Client [+ server] authentication	Client [+ server] authentication	Client + server authentication	Client + server authentication + Integrity protection
Transport layer	Server authenticated TLS	Server authenticated WTLS, i.e. c 2	No requirements	No requirements
Bearer layer	No requirements	No requirements	No requirements	BT ciphering and authentication

Table 1 Security mechanisms per protocol layer

From this we can conclude that there are strong requirements for client authentication and integrity protection at SyncML level since there are scenarios where there are no alternatives.

Server authentication and confidentiality are useful at SyncML level but it is not essential.

SyncML specifies an authentication protocol that can be used for both client and server authentication. Recently there have been proposals on how to incorporate integrity protection by adding a new <security> tag.

The main problem with SyncML security is that it is based on a username, password scheme. This has two drawbacks it gives us weak security and it forces the user to handle yet another password. It is also difficult to derive good integrity protection keys from a password.

Summary of the invention

An object of the invention is to provide strong authentication, which also generates good key material for integrity protection.

The object of the invention have been achieved based on four different technologies:

- GSM SIM, Client authentication and good integrity keys
- UMTS USIM, mutual authentication and good integrity keys
- Proprietary authentication token technology such as SecureId, SafeWord etc.
- PKI based schemes, e.g. WPKI and WIM

A similar solution can be found for SIP security in 3GPP. That solution is based on the use of USIM and http authentication/integrity protection.

The detailed solution is found by carrying GSM/UMTS/ authentication data in the existing SyncML authentication protocol. We would also need to devise a new authentication scheme in SyncML to signal what scheme we are using.

Detailed disclosure of embodiments

The message flow shown below illustrates the use of SIM/USIM/etc as the authentication and integrity protection mechanism for SyncML. Similar message flows can also be drawn for other schemes such as PKI or WIM based systems.

The protocol, incorporating interaction with SIM/USIM and AUC is described in the figure below.

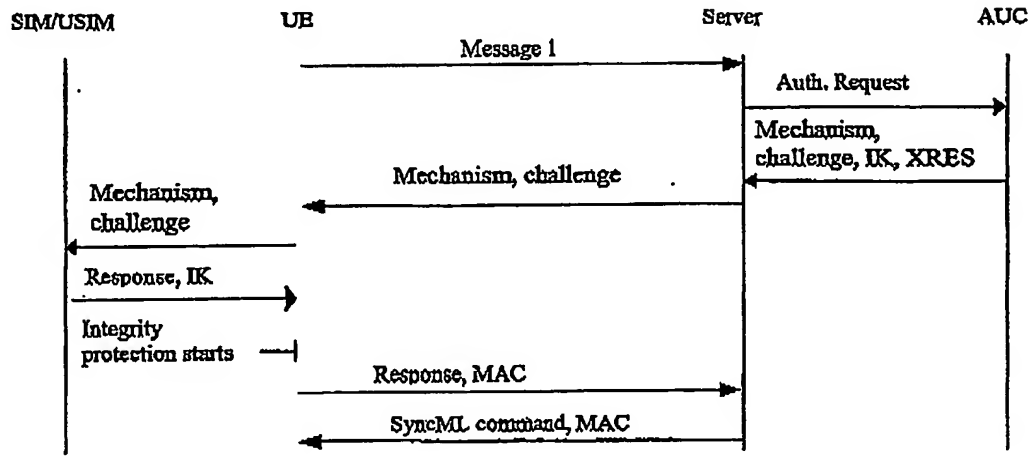

Figure 1. Authentication and integrity protection

These steps are performed:

1. Message 1 sent to server
2. The Server contacts the authentication center, AUC to get a mechanism, challenge, integrity key, Ik and an expected result, XRES
3. AUC replies with said information.
4. The Server sends the mechanism and challenge to the client.
5. The client transfers the challenge to the SIM/USIM, which calculates a response and an integrity key, IK. The mechanism can be used to indicate if the it is a SIM or USIM challenge.
6. The result is sent to the server and integrity protection starts. The Ik is used to generate the MAC.

The main difference between USIM and SIM authentication is that USIM authentication, AKA, provides server authentication in addition to client authentication.

DIGEST CHALLENGE PARAMETERS

If the protocol were implemented in SyncML then the challenge would have the following syntax.

```
<Cred>
   <Meta>
      <Type xmlns="syncml:metinf">syncml: mechanism </Type>
   </Meta>
   <Data 18EA3F...../Data>
<!-- base64 formatted challenge -->
</Cred>
```

The value of the mechanism would in this case be:

mechanism = auth-SIM-HMAC-SHA1 OR: mechanism = auth-md5

The "algorithm" directive is a string that indicates the algorithm(s) used to produce the message digest and an authentication method. The particular strings that may be associated with this directive are not standardized. This contribution illustrates the possible usage of a SHA-1 based HMAC as the digest (MAC generation) algorithm,

Currently, Digest specifies a hash on user name and password to derive the secret key that is used by the MAC generation algorithm. However, it the integrity key IK can be produced either by executing UMTS AKA or by executing the GSM authentication procedure.

The challenge value, generated by the server and sent in the challenge to the client, is used to prevent replay attacks and is implementation dependent. In 3GPP the server can use the equivalent of the AKA FRESH parameter as the challenge value. This value, along with the parameter nonce-count, is used for full anti-replay protection.

Our scheme can be expanded to cover PKI based mechanisms using WIM or other schemes based on alternative authentication token technology by adding new mechanism values.

It should be noted that the same protocol could also be integrated below SyncML in transport protocols such as http or Obex.

DIGEST RESPONSE PARAMETERS

If the protocol were implemented in SyncML then the response could have the following syntax.

```
<Cred>
   <Meta>
      <Type xmlns="syncml:metinf">syncml: mechanism </Type>
   </Meta>
   <Data 1F8CA35……/Data>
<!-- base64 formatted response -->
</Cred>
<SyncSec>
   <Meta>
      <Type xmlns="syncml:metinf">syncml: mechanism </Type>
   </Meta>
      <SyncMAC>123458976736</SyncMAC>
</SyncSec>
```

The value of the "response" is the output from the SIM/USIM. And the value of the MAC is computed as follows

The MAC is computed as per RFC2104, and uses SHA-1 as its hashing function.
The MAC relies upon the use of a shared secret (or key). The key MUST be the integrity key, IK generated by the SIM/USIM.

The HMAC is computed on the entire SyncML DM message. Each SyncML DM message is constructed as normal, upon completion of the message, the HMAC is computed.

Again it should be noted that the same protocol could also be integrated below SyncML in transport protocols such as http or Obex. This is perhaps most applicable to the <SyncSec> tag. To avoid changing the SyncML defenition the same information can be put before SyncML in for example the http body.

MERITS OF INVENTION

- Provides strong authentication based on SIM/USIM and AUC
- Provides integrity protection using good keys derived from the authentication scheme.

- Allows various hash functions and MAC generation algorithms to be used.
- Provides anti-replay protection without the need for synchronized counters in both client and server.

Claims

1. A method for providing authentication of a client adapted to exchange data with a server and integrity protection of data exchanged between the client and the server using a transmission protocol, characterized in that

the server upon reception of a first message from the client, or transmission of a first message to the client, for establishing a connection to the client, transmits an authentication request to an AUC (authentication center);

the server receives a challenge, a mechanism indicating the type of challenge, a first IK (integrity key), and an XRES (expected result) from the AUC;

the server stores the first IK and the XRES, and transmits the mechanism and challenge to the client;

the client receives said challenge and said mechanism and transfers the challenge and the mechanism to a the client, which calculates a response and a second integrity key; and

the response and a MAC generated by the client is transmitted to the server for comparing the response and XRES and starting the integrity protection.

2. The method according to claim 1, wherein the generation of the challenge in the AUC is based on SIM authentication, which is indicated by the mechanism.

3. The method according to claim 1, wherein the generation of the challenge in the AUC is based on USIM authentication, which is indicated by the mechanism, and wherein said mechanism further provides server authentication.

4. The method according to claim 1 or 2, wherein a SIM utilizing SIM authentication for calculating the second IK and the response is provided in the client.

5. The method according to claim 1 or 3, wherein a USIM utilizing USIM authentication for calculating the second IK and the response is provided in the client.

6. The method according to any of the previous claims wherein the MAC is generated based on the second IK.

7. The method according any of the previous claims, wherein the transmission protocol is the SyncML protocol.

8. The method according to any of the claims 1-6 claims, wherein the transmission protocol is the http, WSP or Obex protocol.

9. A method for providing authentication of a client adapted to exchange data with a server and integrity protection of data exchanged between the client and the server using a transmission protocol, characterized in that

the server upon reception of a first message from the client, for establishing a connection, generates a challenge;

the server transmits the challenge to the client;

the client receives from the server said challenge and generates a result based on the challenge;

the response is transmitted to the server, which requests a check by a second server.

10. The method according to claim 9, wherein the generation of the challenge is based on PKI based authentication and integrity protection.

11. The method according to claim 10, wherein the second server is a CA (certificate authority).

12. The method according to any of the claims 9-11, wherein said response comprises an authentication certificate.

13. The method according to claim 9, wherein the generation of challenge is based on SafeWord authentication and integrity protection.

14. The method according to claim 9 or 13, wherein the second server is an AUC (authentication center) server.

15. The method according to any of the claims 9-12, wherein a WIM generating the response is provided in the client.

16. The method according to any of the claims 9 or 13-14, wherein a SIM or USIM is provided for generating said response.

17. The method according any of claims 9-16, wherein the transmission protocol is the SyncML protocol.

18. The method according to any of the claims 9-16, wherein the transmission protocol is the http, WSP or Obex protocol.

19. A client capable of exchanging data with a server according to a transmission protocol adapted to provide authentication and integrity protection, **characterized** in that the client is adapted to

transmit a message for establishing a connection with a server;

receive a challenge and a mechanism, generate a response and integrity key from based on the challenge and

the mechanism, and transmit the integrity key to the server together with a MAC calculated by the client.

20. The client according to claim 19, **characterized** by a smartcard, which generates said response and said integrity key.

21. The client according to claim 20, wherein the smartcard is a SIM or USIM.

22. The client according to any of the claims 19-21, wherein the client is adapted to exchange data according to the SyncML protocol.

23. The client according to any of the claims 19-21, wherein the client is adapted to exchange data according to the http, WSP or Obex protocol.

24. A server capable of exchanging data with a server according to a transmission protocol adapted to provide authentication and integrity protection, **characterized** in that the server is adapted to

receive a message for establishing a connection to a client;

transmit an authentication request;

receive a mechanism, a challenge, an IK and an XRES;

transmit the mechanism and the challenge and store the IK and the XRES; and

receive a response and a MAC; and

compare the XRES and the response.

25. The server according to claim 24, wherein the server is adapted to exchange data according to the SyncML protocol.

26. The server according to claim 24, wherein the server is adapted to exchange data according to the http, WSP or Obex protocol.

27. A method for providing authentication of a client adapted to exchange data with a server and integrity protection of data exchanged between the client and the server using a transmission protocol, characterized in that

the server upon reception of a first message from the client, for establishing a connection with the client, transmits an authentication request to the client;

the client receives said authentication request and generates a password, which is transmitted as a response to the server,

the server receives said response and transmits a request, for a check of said response, to a second server.

28. The method according to claim 27, wherein the second server is an authentication server.

29. The method according to claim 27 or 28, wherein a SIM or USIM is provided in the client for generating the password.

30. The method according to any of the claims 27-29, wherein the authentication and integrity protection is based on SecureId.

31. The method according any of the claims 27-30, wherein the transmission protocol is the SyncML protocol.

32. The method according to any of the claims 27-30, wherein the transmission protocol is the http, WSP or Obex protocol.

Abstract

According to the method of the invention authentication of a client, which is adapted to exchange data with a server, and integrity protection of data exchanged between the client and the server using a transmission protocol is provided. More particularly, the server transmits upon reception of a first message from the client, for establishing a connection, an authentication request to an AUC (authentication center). Then the server receives a challenge, a mechanism indicating the type of challenge, a first IK (integrity key), and an XRES (expected result) from said AUC, stores the first IK and the XRES, and transmits the mechanism and challenge to the client. According to the method of the invention, the client receives said challenge and said mechanism and transfers them to a smartcard of the client, which calculates a response and a second integrity key. Further, the response and a MAC generated by the client is transmitted to the server for comparing the response and XRES and starting the integrity protection.